

Designing Positioning Coordinates of Powered Wheelchairs with Radio Frequency Identification

Chien-Yu Lu,¹ Hui-Fen Wu,¹ Ching-Chou Tai,¹ Kai-Yi Hsueh,¹ and Te-Jen Su,^{2,3*}

¹Department of Industrial Education and Technology, National Changhua University of Education,
Changhua City, Taiwan 500, ROC

²Department of Electronic Engineering, National Kaohsiung University of Science and Technology,
Kaohsiung City, Taiwan 807618, ROC

³Graduate Institute of Clinical Medicine, Kaohsiung Medical University,
Kaohsiung City 80708, Taiwan, ROC

(Received December 30, 2020; accepted March 26, 2021)

Keywords: radio frequency identification, quantum geographic information system, powered wheelchairs, boundary

We combine radio frequency identification (RFID) and the Quantum Geographic Information System (QGIS) to develop a positioning and warning system for powered wheelchairs rented for activities within a specific area. First, we begin with the use of Arduino to drive ultrahigh-frequency (UHF) RFID to read an Electronic Product Code (EPC) tag and take the information of the tag to design an area boundary detection warning function for powered wheelchairs in use. Because the Global Positioning System (GPS) takes more time to start and is greatly affected by the weather and location, we use the relative position of the powered wheelchair to design the positioning system instead of the GPS chip positioning system. By combining the coordinate map of the active area designed by Python and the QGIS, the current position of the wheelchair relative to the point where it was rented is calculated. Finally, the Arduino and QGIS programs are integrated through Python to establish the positioning coordinate system of the powered wheelchair with RFID identification.

1. Introduction

With the advances in medical technology in recent years, human life expectancy has continued to increase, mortality rates have declined, and the world is gradually becoming an aging society. Elderly people and people with medical needs often have mobility issues, including loss of walking ability, poor walking efficiency, safety concerns, cardiopulmonary failure, and musculoskeletal injuries. Alternatively, a person may be unconscious but need to be transported in a sitting position. People with limited mobility often need mobility aids for self-care as well as workplace, school, and leisure activities. Wheelchairs are a widely used means of assisting transportation used by many people with physical impairments and senior citizens in society. Over the past years, wheelchairs have progressed beyond simple mobility aids and have gradually incorporated smart and technological features.

*Corresponding author: e-mail: sutj@nkust.edu.tw
<https://doi.org/10.18494/SAM.2021.3246>

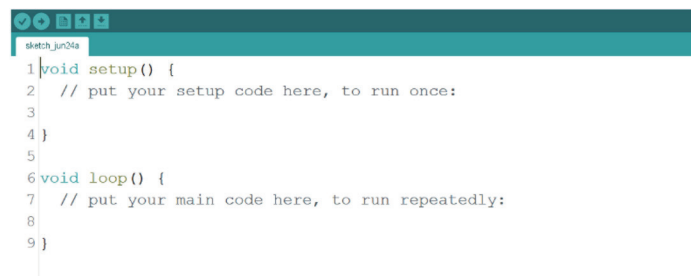
With the changing times and advances in technology, systems using radio frequency identification (RFID) have become ubiquitous; applications include access control,^(1–4) indoor positioning, and object tracking, such as Global Positioning System (GPS) for tracking,⁽⁵⁾ positioning, navigation, item management, security control, guided tours, and disaster relief.

Some event organizers rent out or lend powered wheelchairs in the event area for people with reduced mobility, but are unable to track the wheelchairs when they are rented. This problem has motivated us to design a warning system to track rented powered wheelchairs in public spaces within the event area to prevent them from being removed from the area.^(6–10)

We utilized Arduino, the Quantum Geographic Information System (QGIS), and Python as the software in this study.^(11,12) Arduino was used to drive high-frequency RFID, QGIS was used to design the topographic map of the event area, and Python was used to calculate the relative coordinates. The communication between the main control board of the powered wheelchair, Raspberry Pi 3, and Arduino Uno was then integrated via Python.⁽¹³⁾

Arduino is a single-board microcontroller kit designed and manufactured by Arduino, a company that produces open-source hardware and software. Arduino was developed to provide an economical and easy-to-operate method of creating devices that can connect sensors to any environment for beginner and professional programmers alike, so Arduino allows anyone to make Arduino hardware and software. Arduino also provides the Integrated Development Environment (IDE) editor, as shown in Fig. 1, which is a cross-platform application software in Java programming language. The IDE editor originated from the integrated development environment of the Wiring project and the Processing programming language. This software is designed to introduce programming syntax to people unfamiliar with programming and includes features such as a straightforward syntax, bracket matching, automatic indentation, and one-click compilation. The IDE editor uses a programming language similar to C and C++ and has a software function library containing common input/output functions. After compiling and linking with the GNU toolchain, the IDE editor offers the “avrdude” program to convert the executable file into firmware that can be written into the Arduino hardware.

QGIS is open-source software with the functions of data display, editing, and analysis. Developed by Gary Sherman in 2002,⁽¹⁴⁾ it became a project of the Open-Source Geospatial Foundation in 2004. QGIS is written in C++ programming language and allows plug-in software programming in C++ or Python. Its graphical user interface (GUI) uses a cross-platform C++

The image shows a screenshot of the Arduino IDE software interface. At the top, there is a dark teal header bar with several icons on the left. Below the header, the main area is a code editor with a white background. The code is written in a monospaced font and is as follows:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

Fig. 1. (Color online) Design interface of Arduino IDE software.

application development framework (Qt), Geometry Engine Open Source (GEOS), and a relational database management system (SQLite). Additionally, it supports the Geospatial Data Abstraction Library (GDAL), the Geographic Resources Analysis Support System (GRASS) GIS, PostGIS, and an open-source object-relational database management system (PostgreSQL). QGIS is multi-platform software that runs on a variety of operating systems, including Microsoft Windows, Linux, MacOS X, and UNIX. Compared with the interfaces of other systems, the QGIS interface is faster and more concise. QGIS files are compact and require less memory and processing power than commercial GIS files. Therefore, QGIS can be executed in an environment with limited CPU power or on old hardware.

Python is a widely used general-purpose, interpreting, and high-level programming language. Although it has many object-oriented features, programming in an object-oriented way is not required. The goal of Python is to produce highly readable codes, so its design employs many punctuation marks and uses as many English words as possible often seen in other programming languages to make the codes look simple. Because Python is a dynamic language, unlike static languages such as C and Java, which require statements, its features can be programmed with shorter code. Since its development, it has accumulated a fairly complete standard library (module) and countless non-standard modules, most of which are open-source. The editor used to write Python in this study was Thonny.⁽¹⁵⁾ It has a built-in debugger that provides assistance when encountering errors. It also evaluates the value of regular expressions step-by-step as one of its outstanding features.

Wheelchairs are the most widely used mobility aid as permanent or temporary alternatives to walking. Common users include those with loss of walking ability, poor walking efficiency, safety concerns, cardiopulmonary failure, and musculoskeletal injuries, as well as unconscious people in need of seated transportation.

RFID is an identification technology based on radio frequency, and its principle is shown in Fig. 2. The current signal flows through the antenna coil in the reader, generating a magnetic field at the reader's antenna due to the electromagnetic effect. As the reader approaches the RFID tag, there are changes in the magnetic field, and the antenna coil in the tag generates a current signal to which the microchip in the tag responds before the reader decodes it.

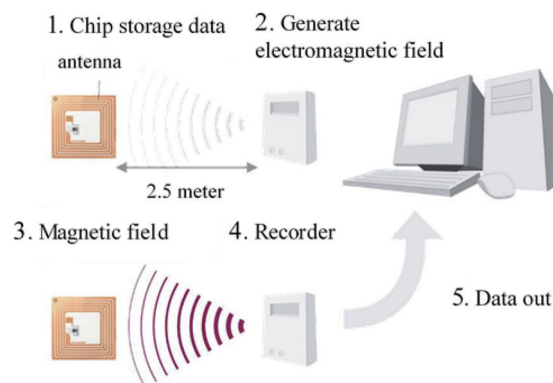


Fig. 2. (Color online) Basic principle of RFID.

RFID was developed in 1939 and originated from British military technology to identify hostile and friendly aircraft during World War II. At that time, the identification devices on fighter aircraft were called transponders, whose functions were similar to that of a tag with RFID architecture, whereas interrogators, analogous to RFID readers, were used on the ground. The area of the RFID system where tags exchange data with the reader is called the interrogation zone and was also derived from military terminology. RFID has a wide range of applications and is most commonly used in identification and retail anti-theft systems, such as EasyCard and access control cards. RFID can replace existing identification tags and product barcodes, and be further integrated with database management systems such as those for controlling incoming and outgoing factory goods, warehouse management, tracking and monitoring, and automated production.

2. System Architecture

RFID systems are now assimilated into our lives, such as EasyCards for mass rapid transit (MRT), anti-theft chips in retail stores or libraries, chips implanted in pets, and MIFARE cards for access control. Many events provide rental powered wheelchairs for people with reduced mobility, but the organizers are unable to track the rented wheelchairs. This problem motivated us to design a warning system to track rented powered wheelchairs in public spaces within an event area to prevent them from being taken out of the area. However, since most wheelchairs are standardized, it is impossible to include more positioning chips in universal powered wheelchairs for positioning design. Also, owing to the fixed range of movement, we did not adopt sensor designs with GPS, ultrasound, and infrared light, but rather utilized the relative positions of the wheelchairs within the allowed area for positioning.

In our system, a powered wheelchair was controlled by Raspberry Pi 3 and UHF RFID technology to establish a boundary warning system for rented powered wheelchairs in an event area. We also created a coordinate map of the wheelchair movement area with QGIS that, in combination with the map, builds a positioning system for event organizers to locate the current positions of wheelchairs, so as to prevent them from being taken out of the event area, as shown in Fig. 3.

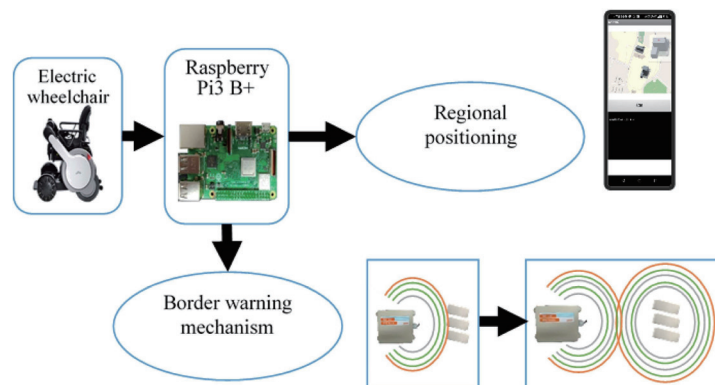


Fig. 3. (Color online) Flow chart of research structure.

The action flow chart introduces the system operation mode of designing the positioning coordinates of the powered wheelchair with RFID, as shown in Fig. 4. One can easily track the condition of a wheelchair from when it is on standby at the rental point to when it is rented and returned.

Raspberry Pi has successively launched Linux-based single-chip computers since 2015. Raspberry Pi 3 is the final version of the Raspberry Pi 3 series released in 2018, as shown in Fig. 5.

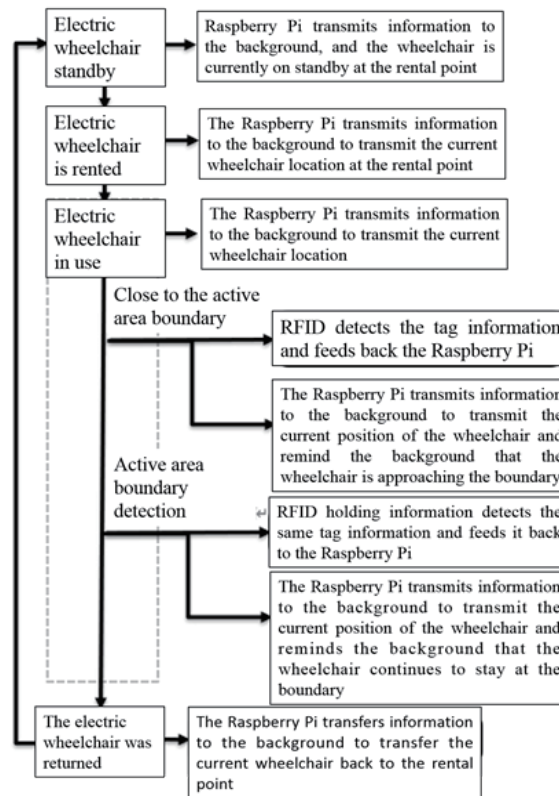


Fig. 4. Action flow chart of designing the positioning coordinates of a powered wheelchair with RFID.

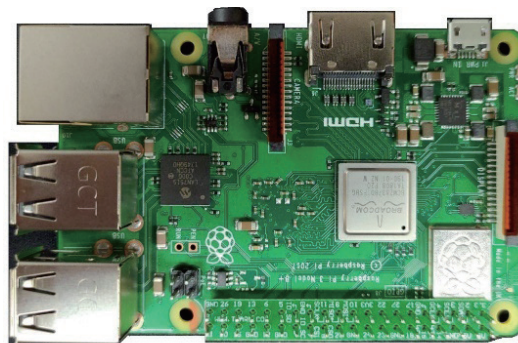


Fig. 5. (Color online) Raspberry Pi 3.

Raspberry Pi is equivalent to a small computer; it is not only easy to obtain, but also offers many options in terms of operating systems. In addition to its built-in Bluetooth and Wi-Fi, it is compact, lightweight, and low in power consumption, making it suitable for a constantly moving wheelchair. Raspberry Pi's transmission and data processing capabilities are very important requirements of the proposed system.

ID01 UHF is a remote RFID tag reader module developed by DFRobot, as shown in Fig. 6.⁽¹⁶⁾ It uses the remote read-write module of an IC card with non-contact UHF technology, which can be easily integrated into the current Wi-Fi network. The maximum operating current is 200 mA, the maximum output power is 24 dBm, the effective range is 50 cm (the attached RFID tag can be detected within 30 cm), and it has a response time of less than 10 ms per 8 bytes when reading.

The communication conversion board is shown in Fig. 7. It is a multi-protocol converter composed of multiple interfaces for communication with the controller. The communication conversion board supports the conversion interfaces among USB, TTL, RS232, and RS485, and allows one input and multiple outputs, making the communication between multiple controllers more convenient.

Arduino Uno is a microcontroller board based on ATmega328P. It has 14 digital input/output pins (six of which can be used as PWM outputs), six analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP connector, and a reset button. It contains everything needed to support the microcontroller; it is readied for use by being connected to a computer with a USB cable or powered with an AC–DC adapter or battery. Uno means one in Italian, and was selected as the name of the released version of Arduino software (IDE) 1.0. Version 1.0 of the Uno development board and Arduino software (IDE) is the reference version of Arduino, which has now been developed into a newer version. The Uno board is the first in a series of Universal Serial Bus (USB) Arduino boards and the model for the Arduino platform. Uno is also the most used and best-documented motherboard in the entire Arduino series, as shown in Fig. 8.

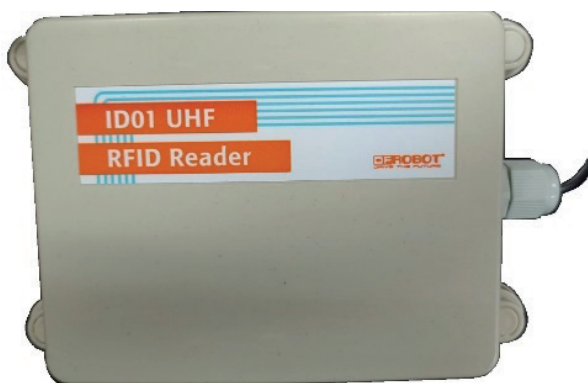


Fig. 6. (Color online) ID01 UHF.

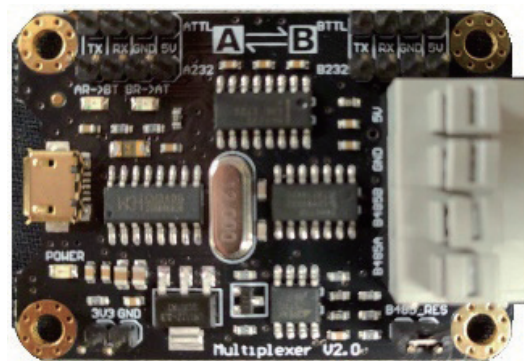


Fig. 7. (Color online) Communication conversion board.

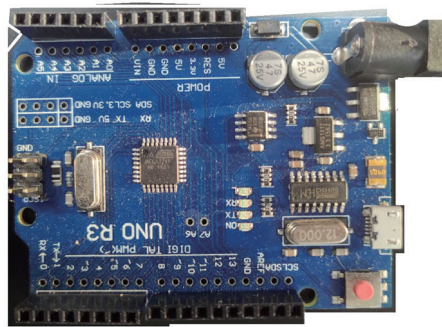


Fig. 8. (Color online) Arduino Uno development board.

3. Experimental Analysis of Designing Positioning Coordinates for Powered Wheelchairs with RFID

First, the pre-system setting of our system is illustrated before describing the system simulation. Then, the research results are discussed in detail for the preparatory system settings of the main core Raspberry Pi, as well as the Arduino and QGIS programming. Because the new Raspberry Pi 3 development board was used, the operating system image file and the Win32DiskImager program had to first be downloaded from the official RPI OS website, and the downloaded and decompressed files were copied to a secure digital (SD) card, which was then connected to Raspberry Pi. After the operating system (OS) was installed, the basic settings were performed by logging into the OS, and various kits were installed as required. After successfully entering Raspberry Pi, the terminal was opened for further domain setting before entering the command to install the Python program, and it was confirmed that the Python development environment was installed. After the completion of the preparatory installation, the Python editor could be opened from the software development option in the Raspberry Pi menu.

We employed Raspberry Pi to design the program to drive the Arduino development board. In addition to acting as the core of the powered wheelchair, Raspberry Pi also buffers the information received and read by RFID, thus preventing the system of the development board from crashing due to transmission overload. To use the Arduino editor in the Raspberry Pi environment, Arduino had to be installed from the terminal. The Arduino Uno development board was used to start UHF ID01, and communication conversion was achieved with the communication conversion board multi-converter. When designing the RFID program, we first introduced the Software Serial library to define a new software serial port, so that signals sent to the Arduino could be received in multiple sequences after RFID tag detection instead of waiting for the signal to be slowly transmitted in the hard serial port. In addition to defining the serial port communication, it also defined the signal transmission rate when the RFID senses a tag to prevent the RFID from failing to transmit read signals, and precluded the entire system from making errors and crashing due to an excessive rate of signal transmission. When reading the EPC tag information for RFID, the bytes and strings required to transfer characters were

defined, and then a string logic operation to declare a write tag instruction was written, which includes the program for RFID tag detection and the logic operation to obtain tag information. The logic operation part that obtained the label information also calculated the operation when there were complex tags. The read serial command was declared to convert the positional notation of the characters read by RFID. In this segment of code, two positional notation conversion types were programmed: the hexadecimal type “HEX” and the decimal type “DEC”, which can be applied accordingly. A debugging command was programmed that determined whether the read tag was the same as the last recorded tag. On the basis of the results of such judgments, either different tags are displayed immediately (the same tags read repeatedly within a period of time are not displayed) or the same tags read repeatedly after a period of time are displayed again. When the RFID read the tag information, the displayed information was annotated. The cell rate was set to 9600 in the initial setting of the Arduino program. In the main Arduino program, the wheelchair returns a warning message of “Alarm! Approaching Boundary!” when sensing a boundary signal.

One can view the tag information and warning messages in each period by clicking on the built-in serial port monitor window of Arduino. After obtaining the information read by RFID, in order for Raspberry Pi to scan the information, one must first return to the Raspberry Pi terminal and enter the command for Python to call the grab serial port menu before using the command to call the narration window of Python, where the cell rate is searched for. The Arduino serial port of the monitor window was called in “Sd.port” and the cell rate was changed in “Sd.baudrate” to enable Python’s connection with Arduino. Commands were entered to call Python’s monitor window to display the information sensed by the boundary. The information monitored by Python was then saved as a separate file, which was sent to the back-end through the cloud. We designed a simulated event map with QGIS software to display an information graph of the location of a wheelchair on the back-end, as shown in Fig. 9, to inform the back-end of the current location of the wheelchair through an information graph.

We simulated a scenario in which a wheelchair was rented, and the current position and boundary warning information of the wheelchair were displayed through the designed interface, as shown in Figs. 10(a)–10(c). Figure 10(a) shows a simulated event area map. In this study, the area surrounding the building on the area map was set as the default boundary, and the sensor

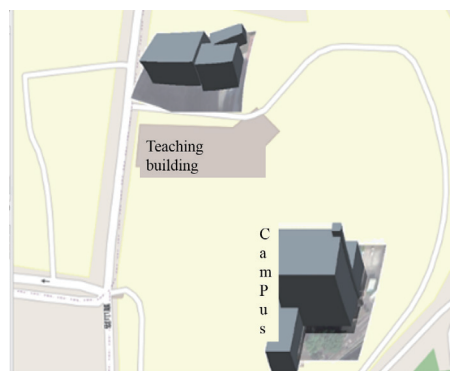


Fig. 9. (Color online) Event area map designed by QGIS.

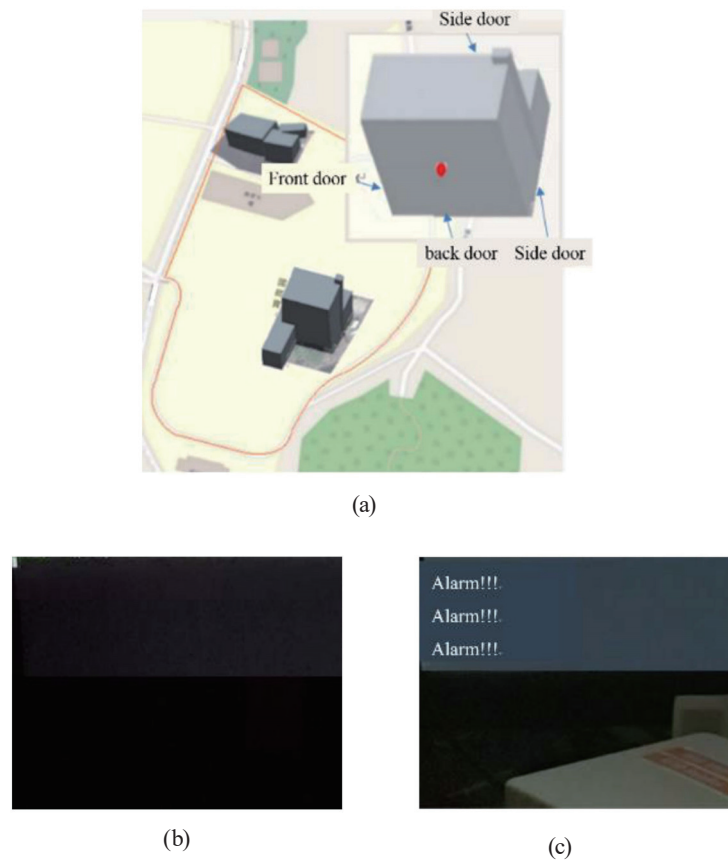


Fig. 10. (Color online) (a) Simulated event area map, (b) tag signal when the boundary is not activated, and (c) warning information for wheelchair at the boundary. The back-end shows the location of the wheelchair.

tags of the RFID system of this study were installed at the entrance and exit of the building (the main entrance, two side doors, and back door). If the wheelchair was within 10 m of the tag, a signal indicating that the wheelchair was near the boundary was sent to the back-end through the cloud every 3.5 ms. The upper right corner of the simulated event map shows the current location of the wheelchair. The wheelchair location in the simulated building is highlighted on this location map. As the wheelchair approached the boundary of the building (the main entrance, two side doors, and back door), the tag signal of the boundary was not activated until the wheelchair reached a distance of 10 m from the boundary, and the back-end displayed the current information interface, as shown in Fig. 10(b). When the wheelchair came within 10 m of the boundary of the building, a tag signal from the preset boundary was received, as shown in Fig. 10(c), and the sensor tag number as well as a warning message that the wheelchair is close to the boundary were displayed.

4. Conclusion

In this study, we developed a powered wheelchair controlled by Raspberry Pi, and an RFID UHF architecture driven by Arduino UNO was integrated into a boundary warning system to

restrict the movement of the wheelchair to a limited area. We first used Arduino to drive UHF RFID to read the EPC tag information, and the read information was used to design the boundary detection and warning function of the wheelchair in use in the area. In addition, QGIS software was used to design a map of the event area, then commands were compiled by Python to build a positioning system with back-end applications for event organizers and manufacturers of powered wheelchairs. This system allows the back-end to track the location of powered wheelchairs that have been rented. Event organizers are able to prevent the theft of wheelchairs through boundary warning and shutdown commands.

References

- 1 F. G. Liu, Y. J. Cai, J. J. Mai, and C. L. Chen: RFID-electronic label, <https://sites.google.com/site/rfiddianzibiaoqian/shou-ye-2> (accessed June 2020).
- 2 D. S. Li: RFID System and EPC Standard Architecture, Master's Thesis (National Taiwan University, Taiwan, R. O. C., 2010).
- 3 N. Wei: RFID Introduction and Application, Master's Thesis (Department of Information Management, National Taiwan University of Science and Technology, Taiwan, R. O. C., 2011).
- 4 Y. H. Xu: J. Hebei North Univ. (Natural Science) **25** (2009) 51 (in Chinese). <https://www.airitilibrary.com/Publication/alDetailedMesh?docid=P20090721002-200904-201001040020-201001040020-51-54>
- 5 Global Positioning System: <https://zh.wikipedia.org/zhtw/%E5%85%A8%E7%90%83%E5%AE%9A%E4%B D%8D%E7%B3%BB%E7%BB%9F> (accessed June 2020).
- 6 Karma Medical: <https://www.karmamedical.com/> (accessed June 2020).
- 7 A. T. Hsieh: Research and Design of Electric Wheelchair, Master's Thesis (Department of Mechanical Engineering, Southern Taiwan University of Science and Technology, Taiwan, R. O. C., 2015).
- 8 Electric Wheelchair Control Interface: <https://www.warmthings.com.tw/blog-post/%E7%89%B9%E6%AE%8 A%E4%BB%8B%E9%9D%A2%E9%9B%BB%E5%8B%95%E8%BC%AA%E6%A4%85%EF%BC%8C%E4 %B8%8D%E6%98%AF%E5%8F%AA%E8%83%BD%E7%94%A8%E6%89%8B%E9%96%8B/> (accessed June 2020).
- 9 M. A. Golovin and A. R. Sufelfa: 2019 IEEE Conf. Russian Young Researchers in Electrical and Electronic Engineering (EIconRus) 514–517.
- 10 B. F. Wu, Y. S. Chen, C. W. Huang, and P. J. Chang: IEEE Access **6** (2018) 28356. <https://doi.org/10.1109/ACCESS.2018.2839729>.
- 11 Wikipedia: <https://zh.wikipedia.org/wiki/QGIS> (accessed June 2020).
- 12 Wikipedia: <https://zh.wikipedia.org/wiki/Python> (accessed June 2020).
- 13 Wikipedia: <https://zh.wikipedia.org/wiki/%E6%A0%91%E8%8E%93%E6%B4%BE> (accessed June 2020).
- 14 Founder of QGIS: Gary Sherman: <https://www.xyht.com/spatial-itgis/godfather-of-qgis/> (accessed June 2020).
- 15 Wikipedia: <https://en.wikipedia.org/wiki/Thonny> (accessed June 2020).
- 16 K. V. Seshagiri, P. V. Nikitin, and S. F. Lam: IEEE Trans. Antennas Propag. **53** (2005) 3870. <https://doi.org/10.1109/TAP.2005.859919>